Cassandra Crossing/ Archivismi: archiviamo Cassandra, parte prima

(564)—Oggi cambiamo lato della medaglia; niente tecnica, raccontiamo una storia vera.

Cassandra Crossing/ Archivismi: archiviamo Cassandra, parte prima
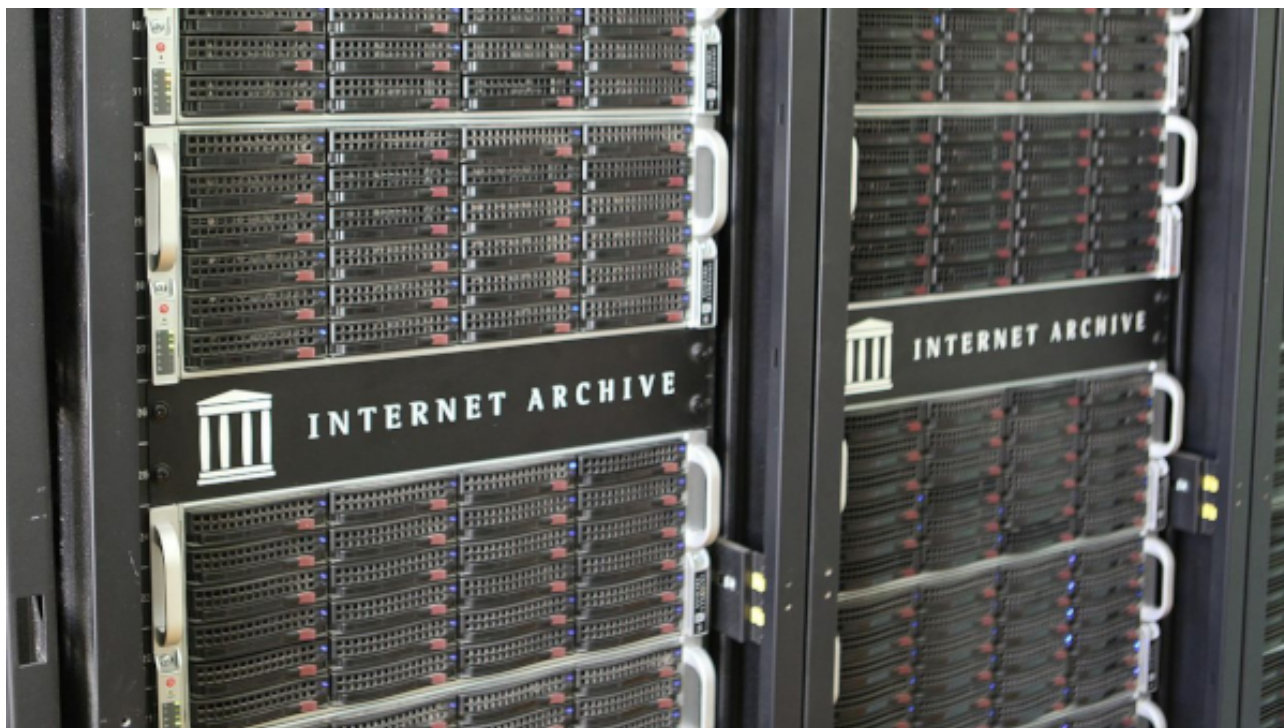
Figure 1:

(564)—Oggi cambiamo lato della medaglia; niente tecnica, raccontiamo una storia vera.

31 dicembre 2023 Nelle ultime tre puntate abbiamo lavorato su Internet Archive, ma solo con esempi semplici.

Archiviare vuole però spesso dire archiviare una quantità di materiali diversi, con uno scopo finale. Ed in questi casi non ci sono esempi semplici che bastino; il diavolo sta sempre nei dettagli, e le informazioni più utili si apprendono ascoltando storie ed esperienze reali.

Ecco che oggi Cassandra vi racconterà una storia vera, tuttora non conclusa, e parlerà solo di dettagli che non hanno a che fare direttamente con Internet Archive, ma con le fasi preliminari una campagna di archiviazione generica, in cui il lavoro più lungo è ritrovare, raccogliere e soprattutto preparare il materiale per l'archiviazione vera e propria.

E cosa di meglio che raccontare la **campagna di archiviazione di Cassandra Crossing**? Si, era da tempo che Cassandra metteva da parte pezzi destinati ad essere archiviati. Ma andiamo con ordine.

Le origini di Cassandra Crossing risalgono al lontano 2003, la pubblicazione regolare (beh, quasi regolare….) inizia invece nel 2005 su Punto Informatico. Prosegue poi su altre testate come Zeusnews.it, talvolta in parallelo. Si estende anche su carta e in video.

I materiali disponibili erano dei tipi più svariati; file di testo con e senza accenti, file di word processor di tipi diversi, file pdf e chi più ne ha più ne metta. Tanti file sono ovviamente andati semplicemente persi.

Fu così che parecchi anni fa Cassandra cercò il modo di recuperare, omogeneizzare e *centralizzare* tutto il *corpus* di Cassandra.

Come in tutte le cose, conviene buttarsi a capofitto in un lavoro, mapensare, programmare, fare e poi cercare una via ancora migliore. Dopo diversi tentativi, Cassandra ha provato Medium.com, un *social specializzato* per scrittori od aspiranti tali. Oltre a fornire un punto unico, in cui scrivere con un discreto editor online ed immagazzinare gli articoli, Medium.con è dotato di una ottima funzionalità di importazione di testo da qualunque sito, anche con pagine piene di pubblicità od effetti vari.

E' dotato di una funzionalità di esportazione dei dati utente, che salvava i singoli articoli in formato in HTML.

Fu così che Cassandra *centralizzò* l'archivio su Medium.com, non senza aver dedicato molto tempo a ritrovare, con i motori di ricerca, i link ai vecchi articoli, mai archiviati in locale o comunque *perduti*.

Ma la soluzione non era soddisfacente per vari motivi, a cominciare dal fatto che gli articoli erano in un cloud, e peggio ancora in quello che sostanzialmente era un social, con tutti gli aspetti deleteri che Cassandra odia e vi racconta spesso.

E così Cassandra decise di iniziare ad archiviare Cassandra Crossing su Internet Archive. E visto che si partiva da un archivio completo in formato omogeneo, sembrava dovesse essere una passeggiata. "Madornale errore", come usa dire Jack Slater.

Infatti l'omogeneità necessaria non è solo una questione di formato, ma soprattutto di struttura interna e di omogeneità delle informazioni memorizzate dei file degli articoli.

Partiamo dalla cosa più semplice: i nomi dei file. Ovviamente Medium.com utilizza una sua filosofia, e forma il nome dalla data di pubblicazione (non quella originaria, ma quella su Medium.com), aggiungendo un identificativo binario ed una derivazione del titolo.

Qualcosa tipo

2023–12–29_Cassandra-Crossing—Archivismi—l-organizzazione-dei-documenti-in-Internet-Archive-e83b9e3b9cca.html

Ora, è pur vero che i file si rinominano anche a mano, ma si tratta di un lavoro improbo quando i file sono centinaia o migliaia. Automatizzare diventa indispensabile. Per fortuna in Linux sono disponibili linguaggi di scripting potenti e librerie che hanno del miracoloso.

Si riesce quindi a rinominare abbastanza facilmente i file togliendo, aggiungendo e riordinando informazioni. Paradossalmente la cosa più difficile è stata inserire automaticamente il numero dell'articolo all'inizio del nome del file.

Per fortuna Cassandra, che talvolta è metodica, aveva l'abitudine di scrivere il numero dell'articolo all'inizio del sottotitolo, mettendolo tra parentesi tonde. Con qualche piccola alchimia di espressioni regolari è stato così possibile estrarlo automaticamente ed utilizzarlo per costruire un più "*umano*" nome di file come

562_Cassandra-Crossing—Archivismi—l-organizzazione-dei-documenti-in-Internet-Archive.html

Poi è stato necessario elaborare i file, ripulirli e convertirli in formati bene archiviabili.

Il primo passo necessario è stato ripulire i file html da una immane quantità di tag nascosti, totalmente inutili per definire il testo ma necessari per garantire le funzionalità del sito di Medium.com. Infatti, come tutti i social, Medium.com implementa le funzioni di esportazioni al minimo sindacale richiesto dal (sempre sia lodato) GDPR, e quindi produce dati completi si, ma non adatti per essere facilmente riutilizzati.

La soluzione migliore che Cassandra ha trovato è stata quella di convertire l'html in formato markdown, filtrare delle linee che non contenevano informazioni utili e riconvertirlo nuovamente in html. Questo piccolo miracolo è stato possibile grazie alle librerie di conversione documentale Pandoc, coadiuvate dalle normali utilità unix come grep.

Ora che i file sono ripuliti ed hanno un nome umano sussiste ancora il problema delle immagini incluse nei file. Infatti le immagini non vengono esportate con gli altri dati, e gli url delle immagini puntano tutti ai server di Medium.com, che quindi, malgrado tutto il lavoro fatto, ha ancora *in pugno* una parte importante degli articoli.

E' necessario quindi convertire le immagini remote in immagini inline, dentro lo stesso codice html, codificandole in base64. Questo processo, concettualmente semplice, deve di solito essere svolto a mano per ogni singolo file ed url; per fortuna esiste il modo di farlo automaticamente, tramite il parametro —*self-contained*, aggiunto al comando Pandoc di riscrittura dell'html.

Per l'archiviazione, il formato principale scelto è comunque il pdf, che non ha questo problema perché convertendo l'html in pdf le immagini vengono inserite direttamente nel file.

Per non farsi mancare niente, sempre grazie ai miracoli di Pandoc, Cassandra ha potuto convertire in maniera semplicissima in pdf tutti i formati già prodotti, l'html di partenza, il markdown e l'html semplificato, scegliendo poi il migliore.

Il risultato, per ora, lo trovate qui.

Concludendo, un paio di giornate "piene" di lavoro hanno portato a questo script bash di 39 righe, certamente non ottimale né privo di errori, che qui comunque commenteremo, giusto per rendere l'idea. Capirlo a grandi linee è sufficiente. Ma se vi servisse, riutilizzarlo sarebbe per voi un bel risparmio di tempo.

```
# Procedura per la preparazione all'archiviazione degli articoli
# di Cassandra Crossing
#
# inizializzazioni varie
_base="./tuttocassandra_elaborazione/"
_base2="./posts/"
_base3="./markdown/"
_base4="./temp/"
_base5="./html/"
_base6="./pdf/"
_temp="temp.txt"
#
# cambio directory di lavoro, creazione directory e pulizia file
cd "${_base}"
mkdir markdown html temp pdf
rm ./markdown/* ./html/* ./temp/* ./pdf/*
cd "${_base2}"
rm "${_temp}"
_dfiles="*"
```

```
#
# inizio loop principale
for f in $_dfiles
do
rm “${_temp}”
#
# estrazione del numero dell’articolo
g=‘grep -Eo -m 1 ‘\([0–9]+\)’ $f | tr -d ‘()’‘
g=“000”$g
g=‘echo $g | rev | cut -c 1–3 | rev‘
h=‘echo $f | cut -d ‘_’ -f2- | rev | cut -d ‘-’ -f2-| rev‘
#
# formazione del nuovo nome del file e copia col nuovo nome
i=$g”_“$h
echo”—-> Identifier: $i”
cp $f “../$_base4${i}.html”
#
# conversione in formato markdown, ripulitura e riconversione in html
pandoc -f html -t markdown “../”$_base4$i”.html” > “${_temp}”
grep -v “^:::” “${_temp}” |sed -e ‘s|||g’ > “../”${_base3}$i”.md”
pandoc—self-contained -f markdown -t html “../”${_base3}$i”.md”> “../”${_base5}$i”.html”
pandoc—pdf-engine=xelatex -f markdown -t pdf “../”$_base3$i”.md” >
“../”${_base6}$i”.pdf”
#
# pulizia e fine ciclo
done
rm -rf “${_temp}” “../$_base4”
```

(Se dovete copiare questa procedura, rimettete a posto i doppi apici curvi con quelli normali, gli apici semplici curvi con quelli normali, il segno meno lungo con due segni meno normali. Medium.com non permette di scrivere come si vuole …)

Ed anche per oggi è tutto. *Stay tuned* per la prossima puntata di “*Archivismi*”.

Scrivere a Cassandra—Twitter—Mastodon
Videorubrica “Quattro chiacchiere con Cassandra”
Lo Slog (Static Blog) di Cassandra
L’archivio di Cassandra: scuola, formazione e pensiero

By Marco A. L. Calamari on January 2, 2024.

Canonical link